CHUYÊN ĐỀ TỔ CHỨC DỮ LIỆU

MÔ HÌNH DỮ LIỆU PHÂN CẤP

Giảng viên: VŨ QUỐC HOÀNG
(vqhoang@fit.hcmus.edu.vn)

Nội dung trình bày

- Quan hệ phân cấp
- Cây
- Cây nhị phân
- Các thao tác trên cây
- Cây nhị phân tìm kiếm

Quan hệ phân cấp (Hierarchy)

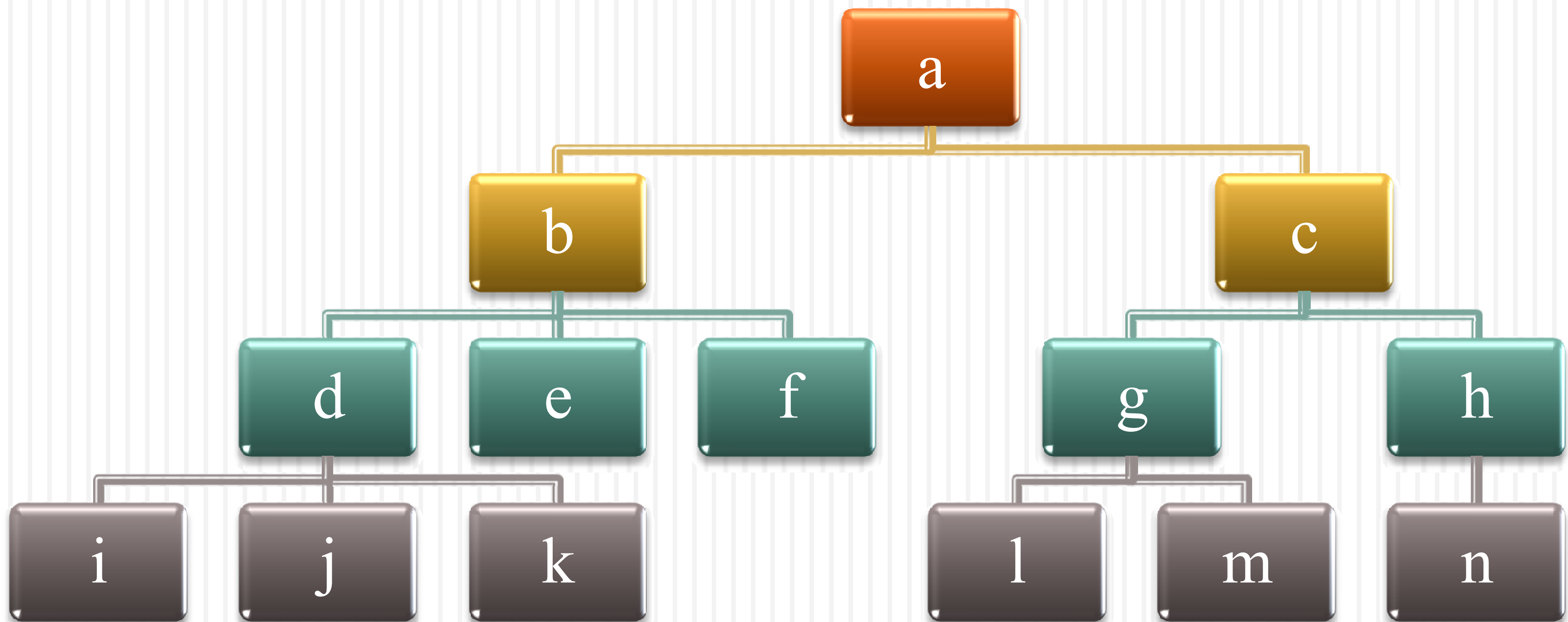
- Quan hệ phân cấp là quan hệ “trên-dưới” về cấp bậc giữa các đối tượng
  - Độ phân cấp: số cấp bậc
  - Độ phân nhánh: số cấp dưới của một đối tượng
- Hai loại quan hệ phân cấp:
  - Quan hệ phân cấp thuần: Một đối tượng có tối đa một cấp trên
  - Quan hệ phân cấp chéo: Một đối tượng có nhiều hơn một cấp trên

Quan hệ phân cấp

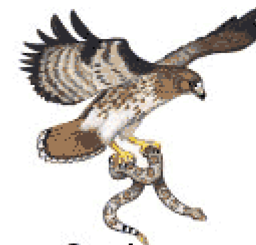
- Các dạng của quan hệ phân cấp:
  - lồng chứa
  - tổ chức
  - kế thừa
  - tiến hóa
  - …

Cây

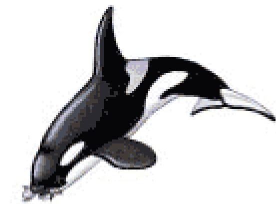
- Cây là quan hệ phân cấp thuần, tức là, mỗi đối tượng chỉ là cấp dưới của tối đa một cấp trên





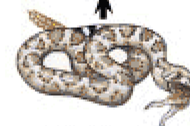


Quaternary consumers

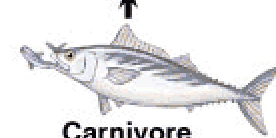


Carnivore

Carnivore

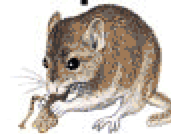


Tertiary consumers

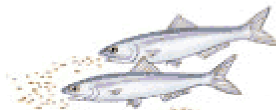


Carnivore

Carnivore

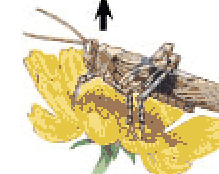


Secondary consumers



Carnivore

Carnivore



Primary consumers

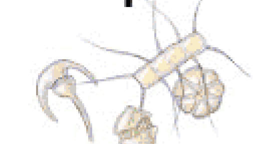


Herbivore

Zooplankton



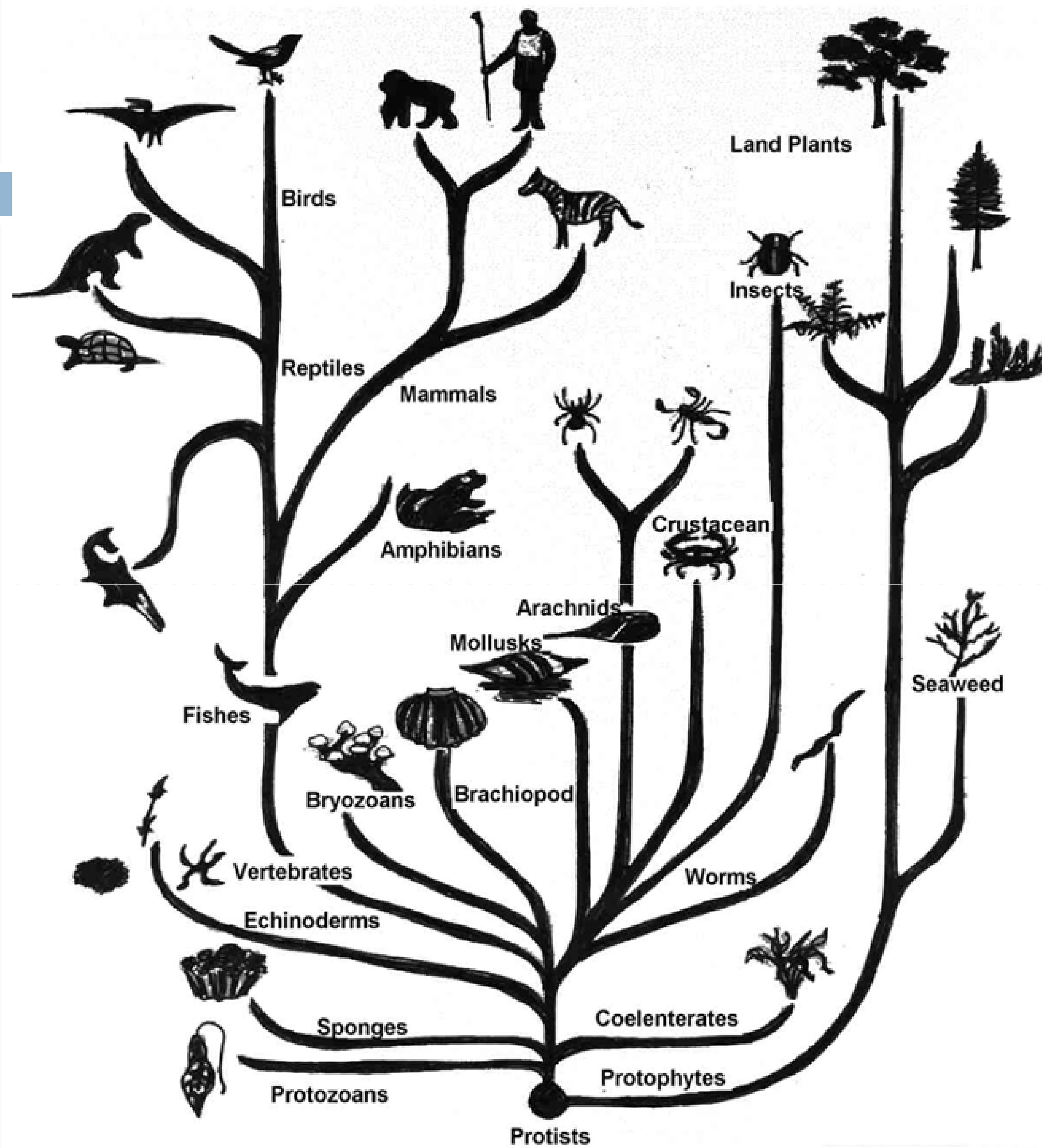
Primary producers

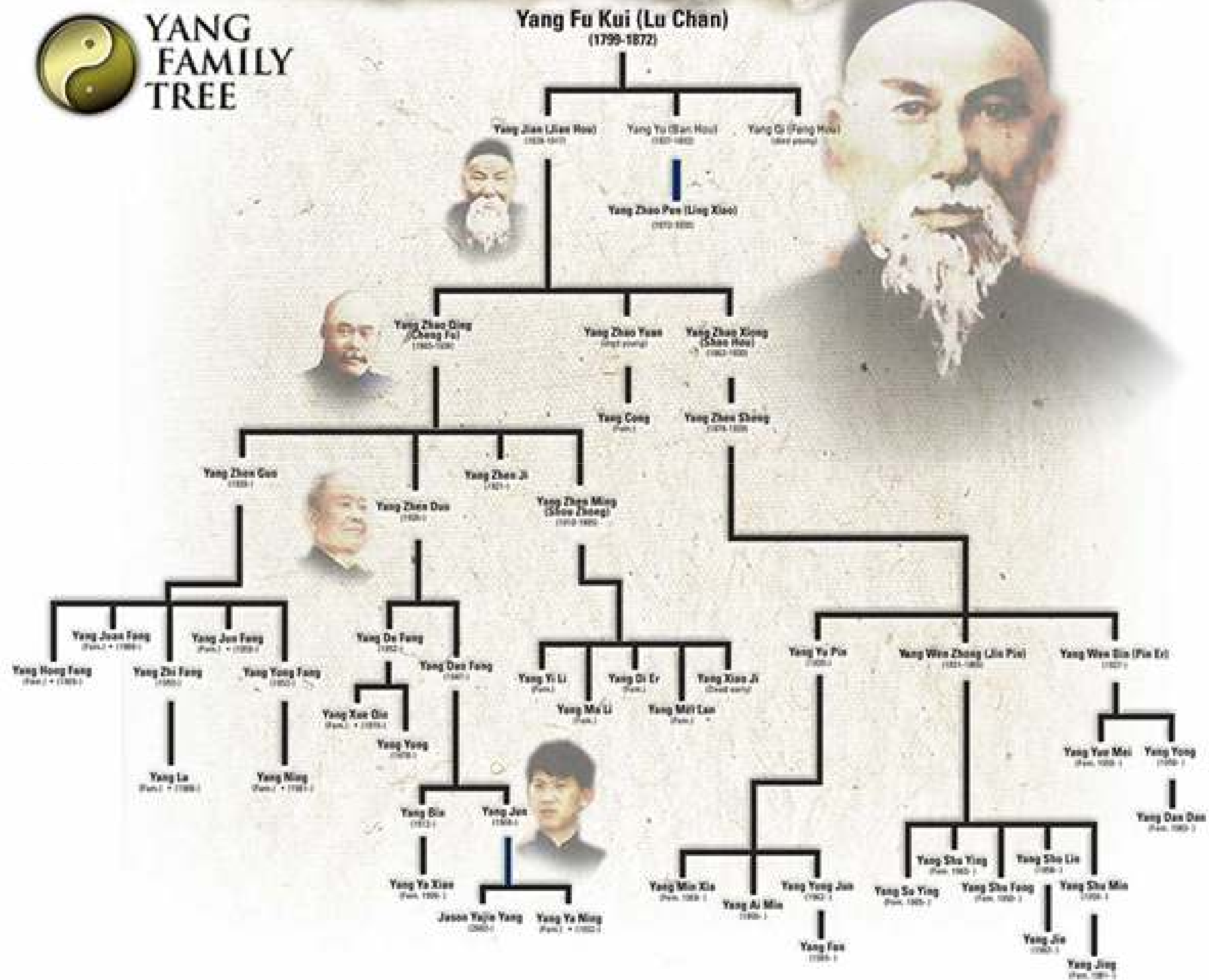


Plant

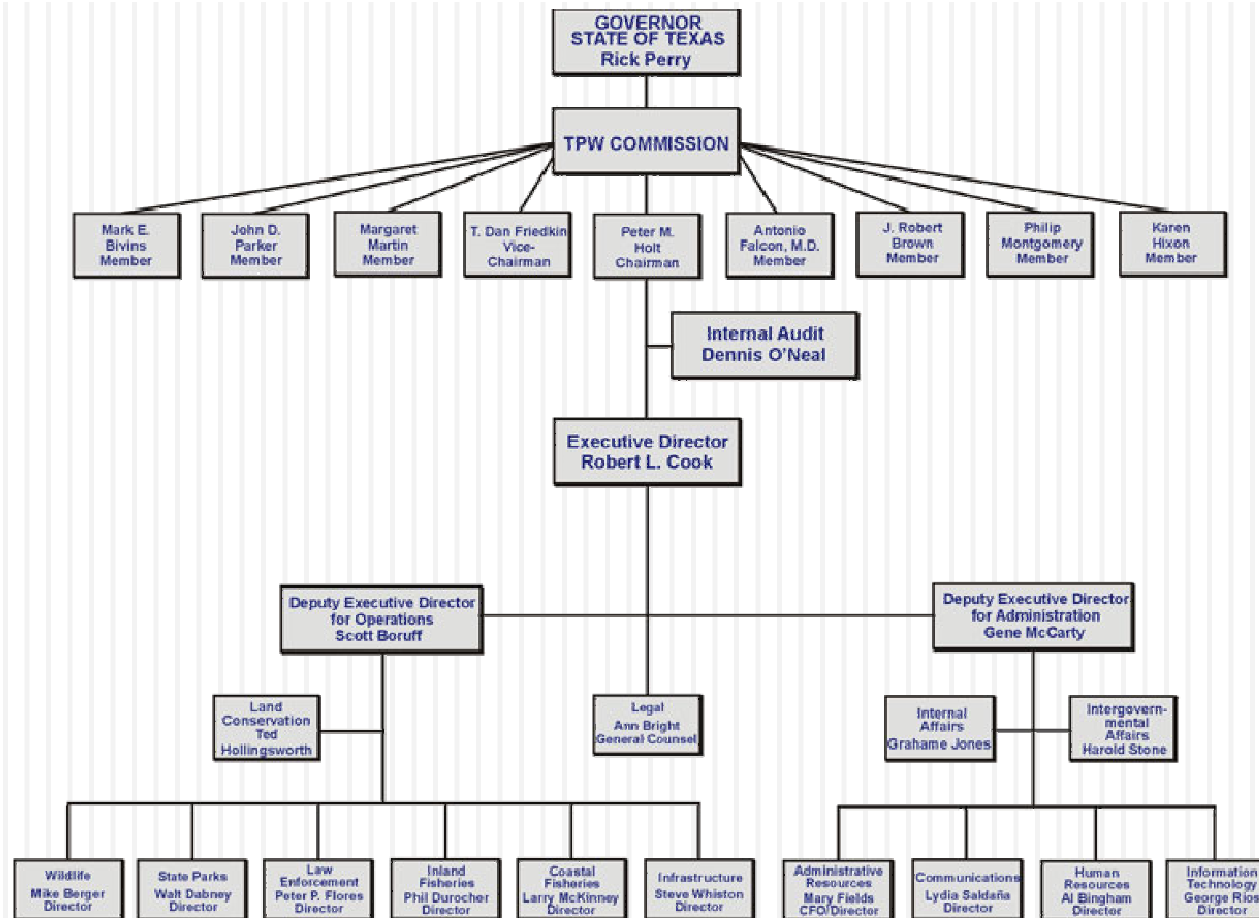
Phytoplankton

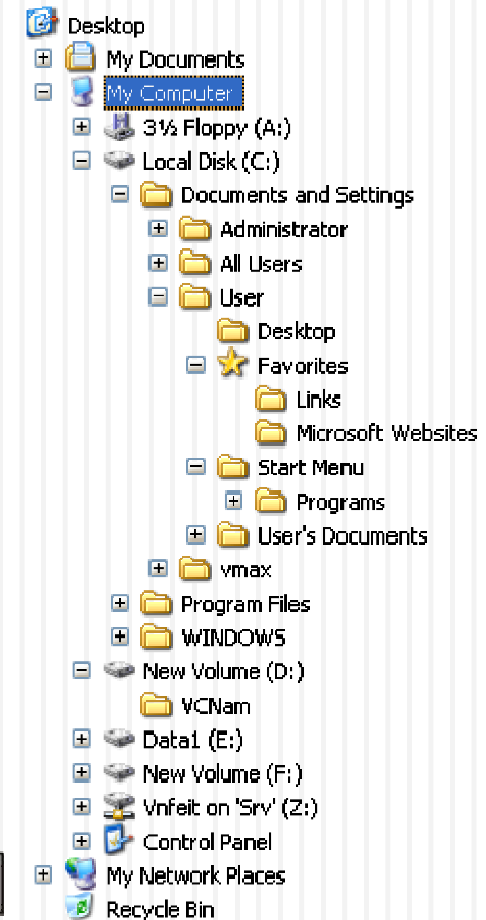
A terrestrial food chain **A marine food chain**


Birds
Land Plants
Insects
Reptiles
Mammals
Amphibians
Crustacean
Arachnids
Mollusks
Seaweed
Fishes
Bryozoans
Brachiopod
Vertebrates
Worms
Echinoderms
Sponges
Coelenterates
Protophytes
Protozoans
Protists



YANG FAMILY TREE
Yang Fu Kui (Lu Chan)
(1799-1872)
Yang Jian (Jian Hou)
Yang Yu (Ban Hou)
Yang Qi (Feng Hou)
Yang Zhao Peng (Ling Xiao)
Yang Zhao Qing (Cheng Fu)
Yang Zhao Yuan
Yang Zhao Xiong (Shao Hou)
Yang Cong
Yang Zhen Sheng
Yang Zhen Guo
Yang Zhen Duo
Yang Zhen Ji
Yang Zhen Ming (Shou Zhong)
Yang Juan Fang
Yang Jun Fang
Yang Hong Fang
Yang Zhi Fang
Yang Yong Fang
Yang De Fang
Yang Dan Fang
Yang Xue Qin
Yang Yong
Yang Lu
Yang Ning
Yang Bin
Yang Jun
Yang Ya Xuan
Jason Yujie Yang
Yang Ya Ning
Yang Yi Li
Yang Di Er
Yang Xiao Ji
Yang Ma Li
Yang Mill Lan
Yang Yu Pin
Yang Wen Zhong (Jin Pin)
Yang Wen Bin (Pin Er)
Yang Yue Mei
Yang Yong
Yang Dan Dan
Yang Min Xia
Yang Ai Min
Yang Yong Jun
Yang Fan
Yang Shu Ying
Yang Shu Lin
Yang Su Ying
Yang Shu Fang
Yang Shu Min
Yang Jin
Yang Jing


GOVERNOR
STATE OF TEXAS
Rick Perry
TPW COMMISSION
Mark E. Bivins Member
John D. Parker Member
Margaret Martin Member
T. Dan Friedkin Vice-Chairman
Peter M. Holt Chairman
Antonio Falcon, M.D. Member
J. Robert Brown Member
Philip Montgomery Member
Karen Hixon Member
Internal Audit
Dennis O'Neal
Executive Director
Robert L. Cook
Deputy Executive Director for Operations Scott Boruff
Deputy Executive Director for Administration Gene McCarty
Land Conservation Ted Hollingsworth
Legal Ann Bright General Counsel
Internal Affairs Grahame Jones
Intergovernmental Affairs Harold Stone
Wildlife Mike Berger Director
State Parks Walt Dabney Director
Law Enforcement Peter P. Flores Director
Inland Fisheries Phil Durocher Director
Coastal Fisheries Larry McKinney Director
Infrastructure Steve Whiston Director
Administrative Resources Mary Fields CFO/Director
Communications Lydia Saldaña Director
Human Resources Al Bingham Director
Information Technology George Rios Director


Desktop
My Documents
My Computer
3½ Floppy (A:)
Local Disk (C:)
Documents and Settings
Administrator
All Users
User
Desktop
Favorites
Links
Microsoft Websites
Start Menu
Programs
User's Documents
vmax
Program Files
WINDOWS
New Volume (D:)
VCNam
Data1 (E:)
New Volume (F:)
Vnfeit on 'Srv' (Z:)
Control Panel
My Network Places
Recycle Bin

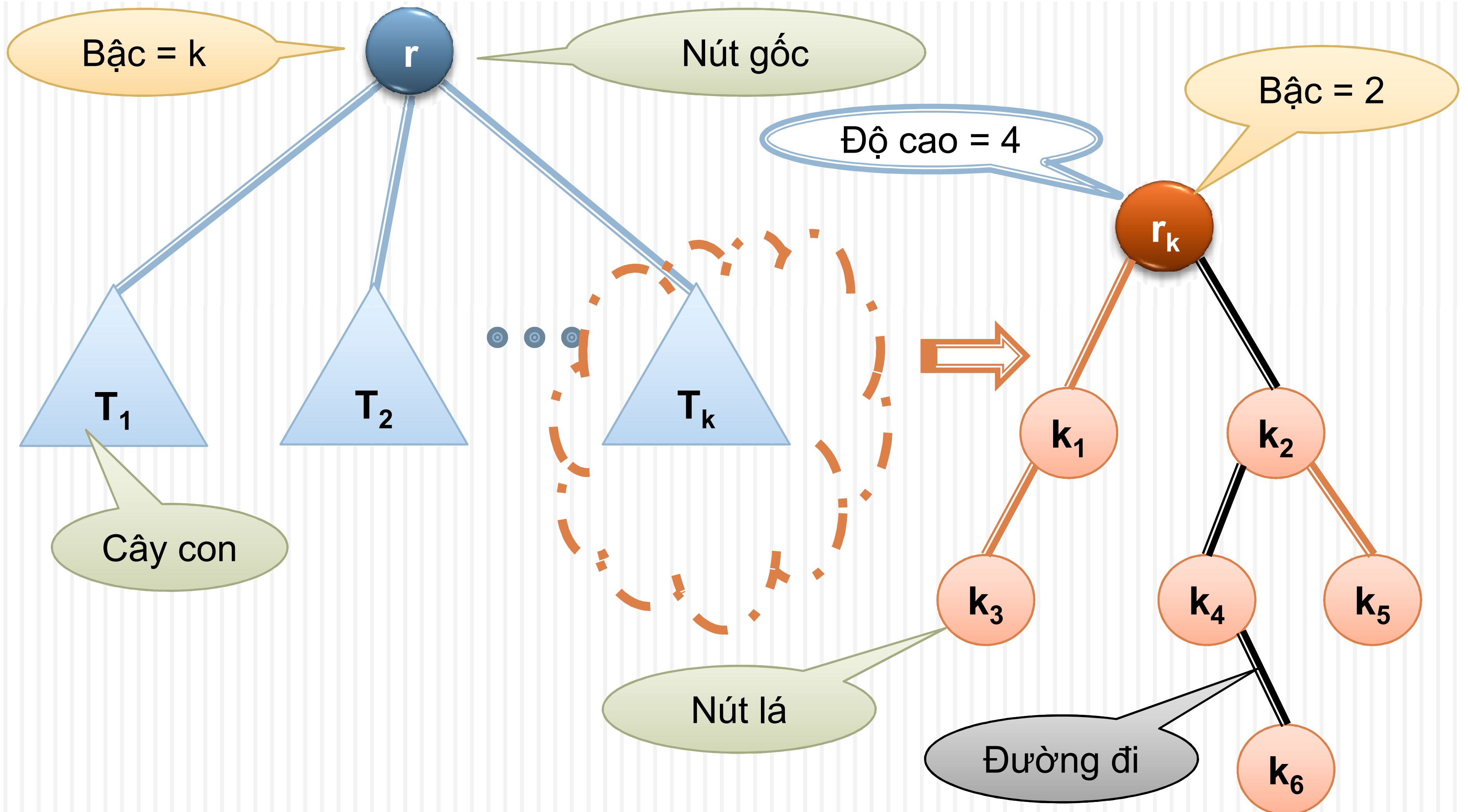
Cây
Các khái niệm

- node: nút
- root: gốc cây
- leaf: lá
- inner node/internal node: nút trong
- parent: nút cha
- child: nút con
- path: đường đi

Cây
Các khái niệm

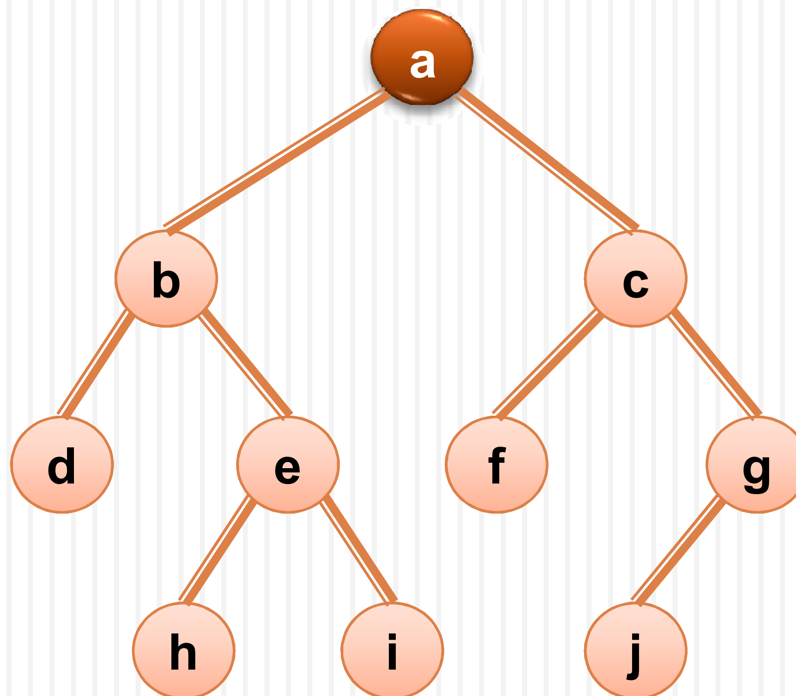
- degree/order: bậc
  - Bậc của node: số con của node
  - Bậc của cây: bậc lớn nhất trong số các con
- depth/level: độ sâu/mức
  - Mức (độ sâu) của node: Chiều dài của đường đi từ node gốc đến node đó cộng thêm 1
- height: chiều cao
  - Chiều cao cây:
    - Cây rỗng: 0
    - Cây khác rỗng: Mức lớn nhất giữa các node của cây

Cây
Các khái niệm

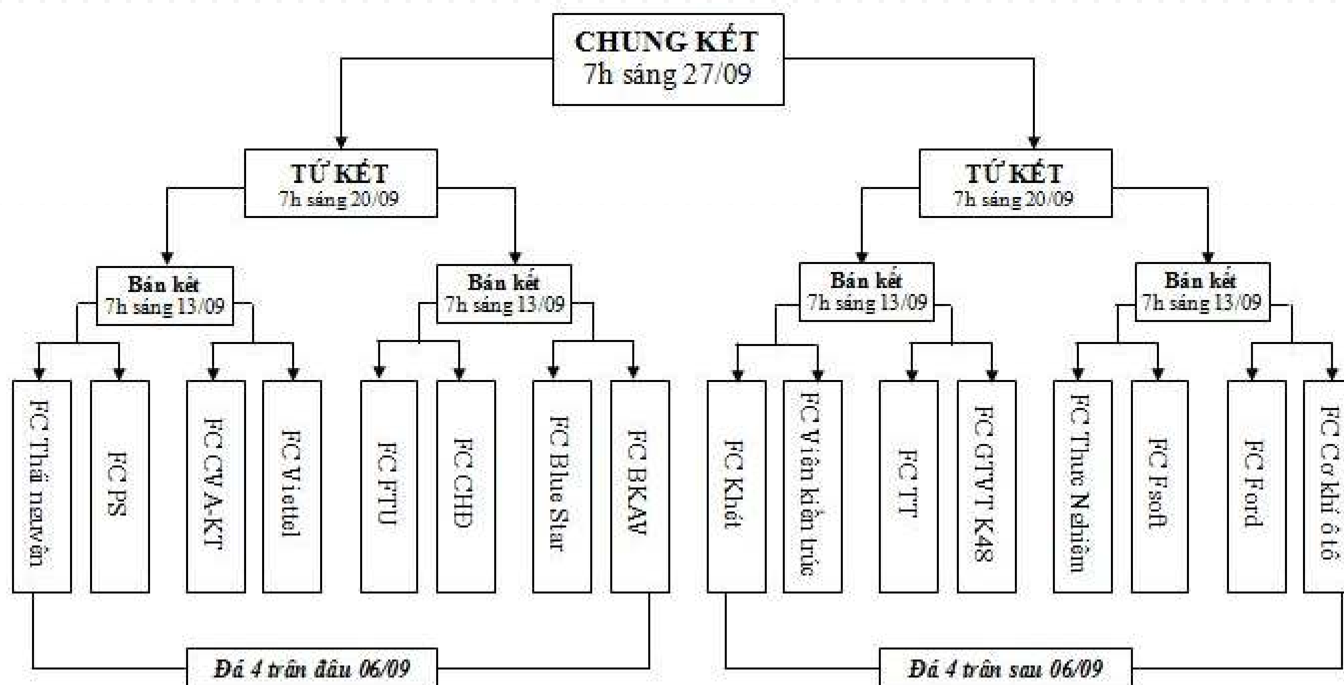


Cây nhị phân

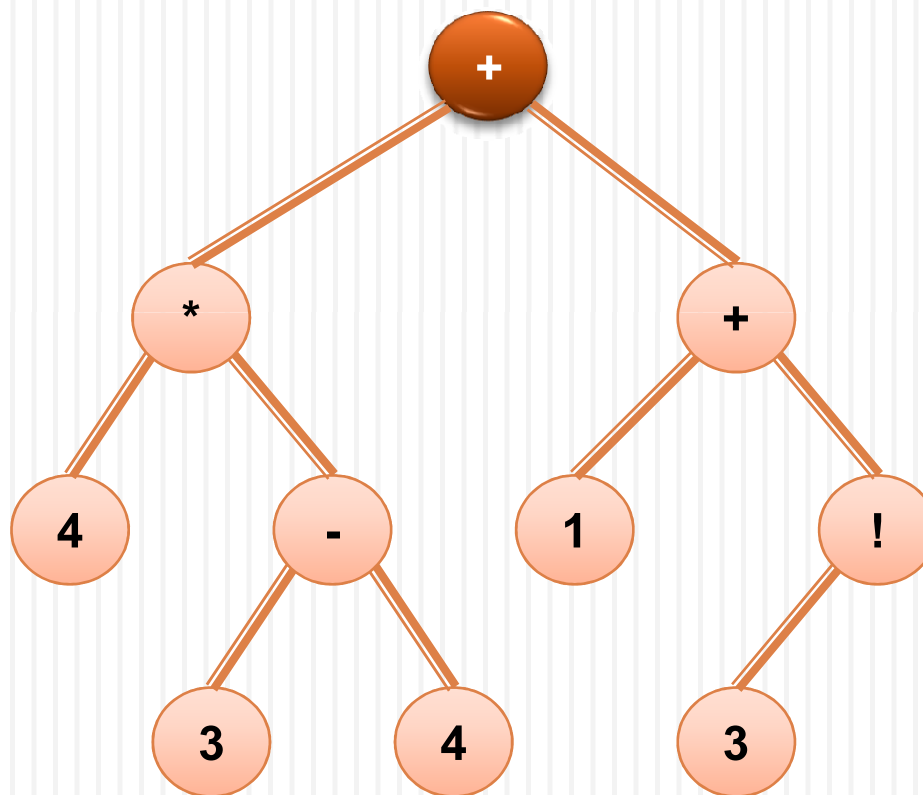
- Cây nhị phân là cây mà mỗi nút có tối đa 2 cây con và có để ý đến thứ tự của các cây con
  - Các cây con được gọi là cây con trái và cây con phải



Cây nhị phân
Cây tổ chức thi đấu



Cây nhị phân
Cây biểu thức số học

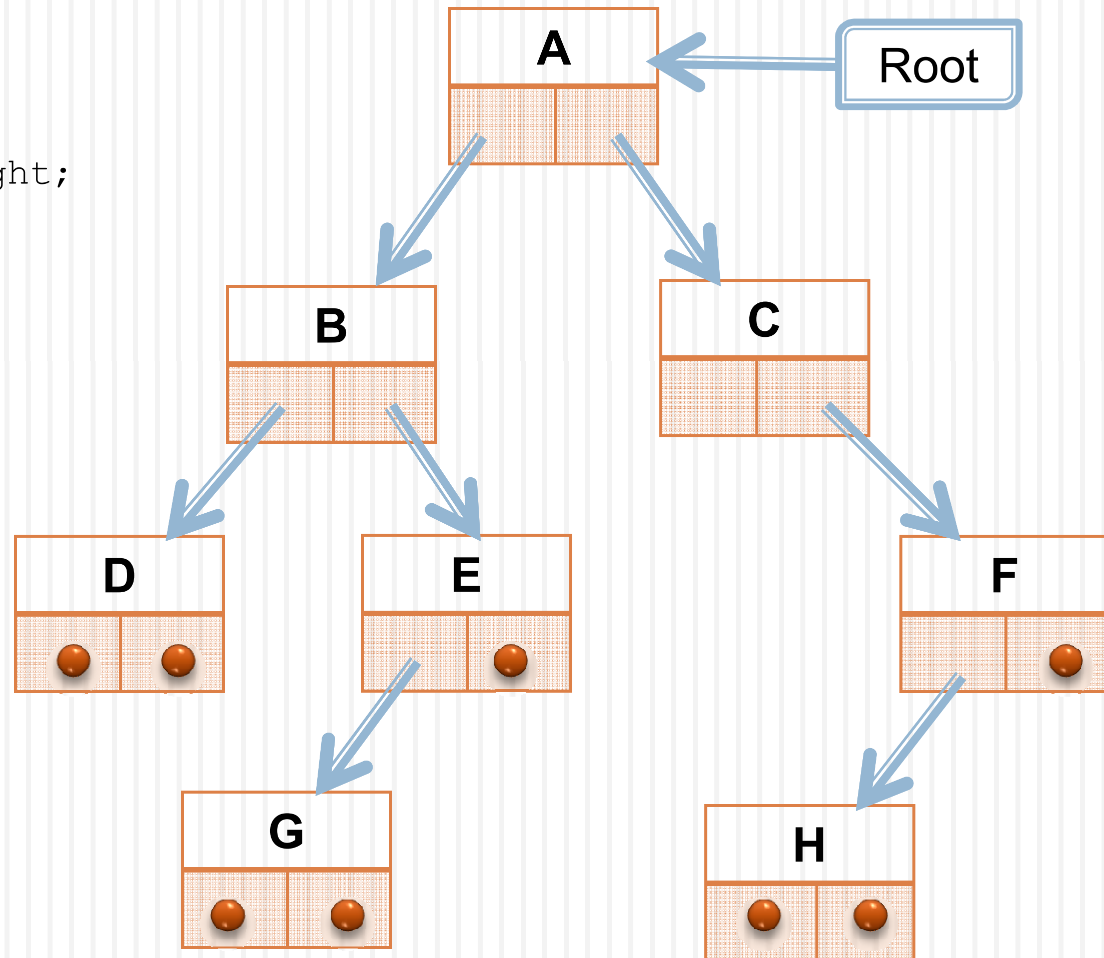


4 * (3 – 4) + (1 + 3!))

Cây nhị phân
Biểu diễn cây bằng các liên kết cây con

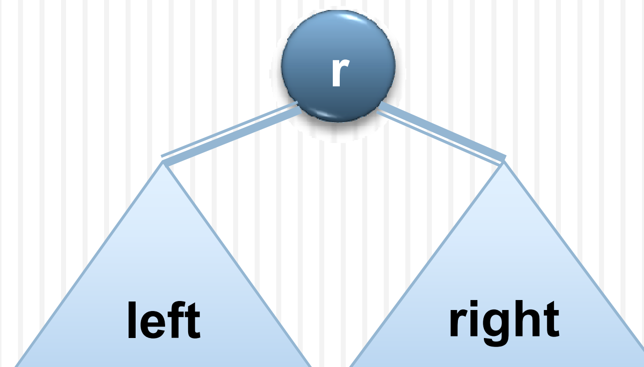

```
struct NODE
{
      DATATYPE data;
      NODE *left, *right;
};

struct BTREE
{
      NODE *root;
};
```



Cây nhị phân

- Cây nhị phân là dạng cấu trúc đệ qui điển hình:
  - *Cơ sở*: cây rỗng là cây không chứa phần tử nào, kí hiệu •
  - *Đệ qui*: với *r* là một phần tử và *left*, *right* là hai cây nhị phân thì có thể tạo một cây nhị phân có gốc là *r*, cây con trái là *left* và cây con phải là *right*



Các thao tác trên cây

- Các thao tác cơ bản trên cây:
  - Duyệt cây
  - Thêm phần tử
  - Xóa phần tử
  - Tìm kiếm phần tử

Các thao tác trên cây

- Vì cây là một cấu trúc đệ qui nên, một cách tự nhiên, các thao tác trên cây là các thao tác đệ qui
- Ví dụ, thao tác đếm số nút của một cây nhị phân *T*, *nodes*(*T*), có thể được làm đệ qui như sau:
  - *Cơ sở*: nếu $T = \bullet$ thì

    $$nodes(T) = 0$$

  - *Đệ qui*: nếu $T \neq \bullet$ thì gọi *L*, *R* là hai cây con trái, phải của *T* ta có

    $$nodes(T) = nodes(L) + nodes(R) + 1$$

Các thao tác trên cây
Thao tác duyệt cây

- Duyệt cây là thao tác "đi qua" một cách hệ thống tất cả các nút trên cây, mỗi nút đúng một lần
- Rất nhiều thao tác xử lý trên cây là phép duyệt cây
- Các phép duyệt cây cơ bản:
  - Duyệt sâu (Depth-first): duyệt theo đường
    - Duyệt trước (Pre-order)
    - Duyệt giữa (In-order)
    - Duyệt sau (Post-order)
  - Duyệt rộng (Bread-first): duyệt theo mức

Các thao tác trên cây
Thao tác duyệt cây


```
void preorder(NODE *r)
{
    // Xu ly r->data
    preorder(r->left);
    preorder(r->right);
}

void inorder(NODE *r)
{
    inorder(r->left);
    // Xu ly r->data
    inorder(r->right);
}

void postorder(NODE *r)
{
    postorder(r->left);
    postorder(r->right);
    // Xu ly r->data
}
```

Đệ qui

```
void breadfirst(BTREE *t)
{
    if(t->root != NULL)
    {
        QUEUE *q = createqueue();
        enqueue(q, t->root);
        while(!isempty(q))
        {
            NODE *n = dequeue(q);
            // Xu ly n->data
            if(n->left != NULL)
                enqueue(n->left);
            if(n->right != NULL)
                enqueue(n->right);
        }
    }
}
```

Dùng hàng đợi

Các thao tác trên cây
Thao tác duyệt cây

Duyệt rộng

- *a b c d e f g h i j k*

Duyệt trước

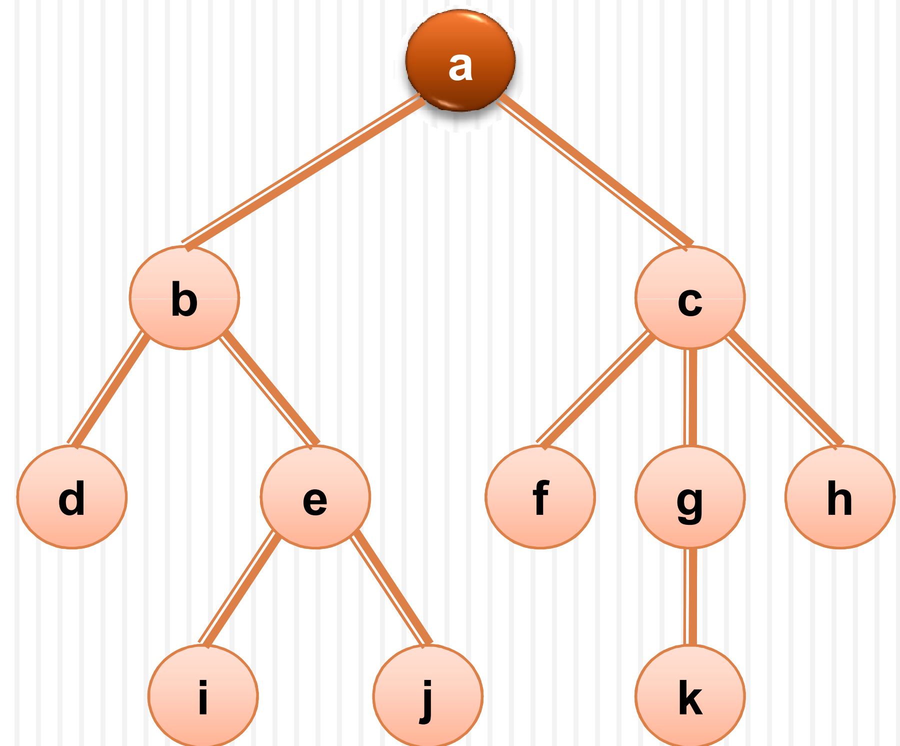
- *a b d e i j c f g k h*

Duyệt giữa

- d b i e j a f c k g h

Duyệt sau

- *d i j e b f k g h c a*



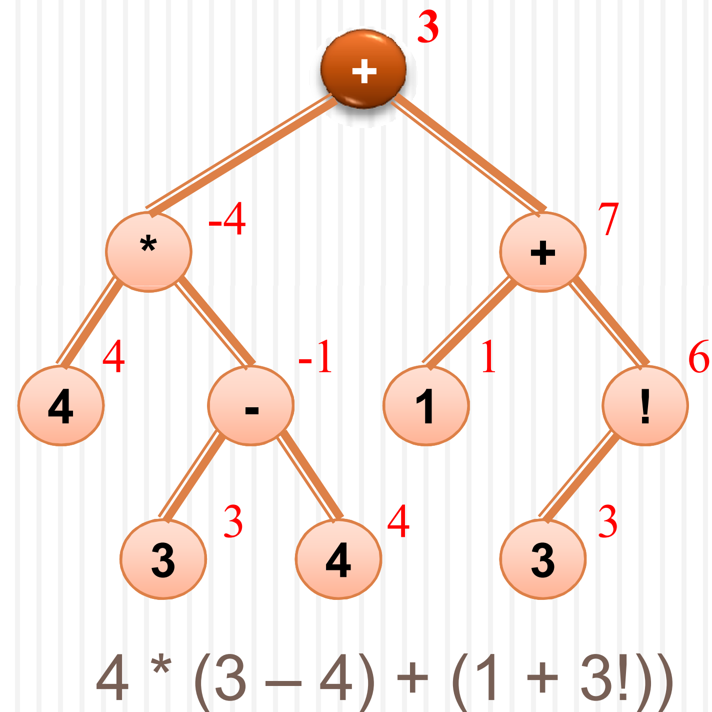
Các thao tác trên cây
Thao tác duyệt cây


```
int eval(NODE *r)
{
    if(r->left == NULL
            && r->right == NULL)
        return r->data;

    if(r->data == '+')
    {
        int v1 = eval(r->left);
        int v2 = eval(r->right);
        return v1 + v2;
    }

    if(r->data == '!')
    {
        int v1 = eval(r->left);
        return factorial(v1);
    }

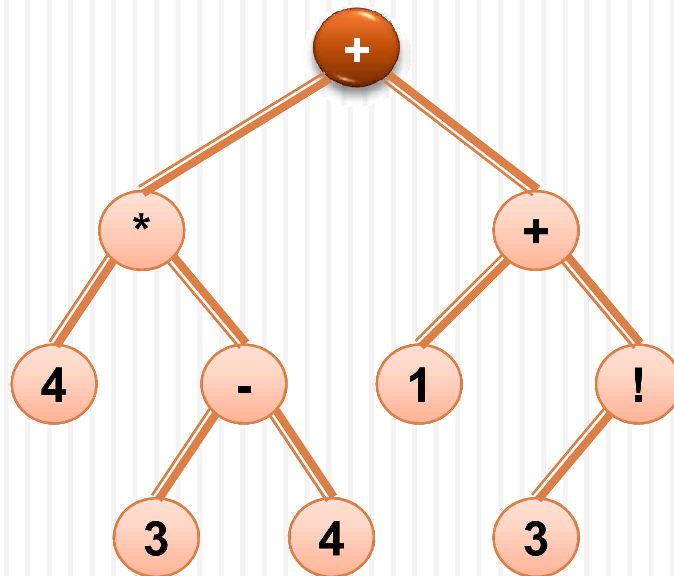
    // ...
}
```

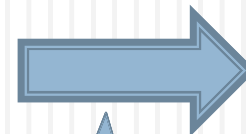


Duyệt sau

Các thao tác trên cây

- Thao tác đưa ra dạng biểu diễn tuyến tính của cây
  - Ví dụ, đưa ra biểu diễn hậu tố của một biểu thức số học từ một cây biểu thức số học





4 3 4 - * 1 3 ! + +

Duyệt sau

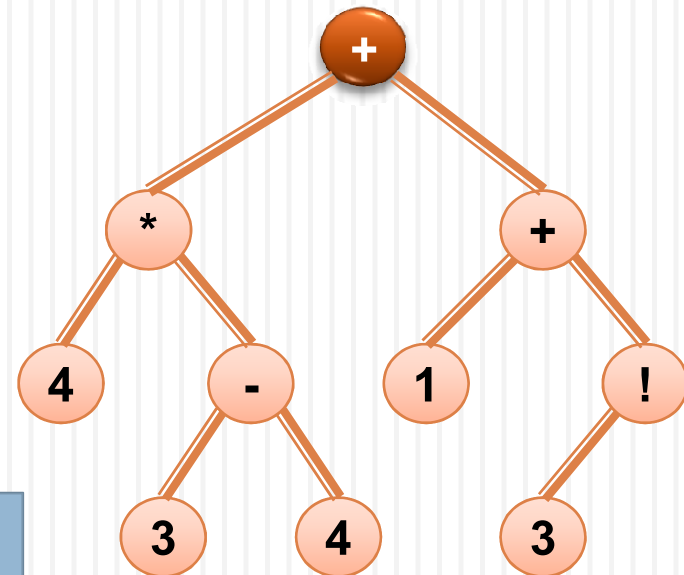
4 * (3 – 4) + (1 + 3!))

Các thao tác trên cây

- Thao tác xây dựng cây từ dạng biểu diễn tuyến tính
  - Ví dụ, xây dựng cây biểu thức số học từ biểu thức số học dạng trung tố

4 * (3 – 4) + (1 + 3!))

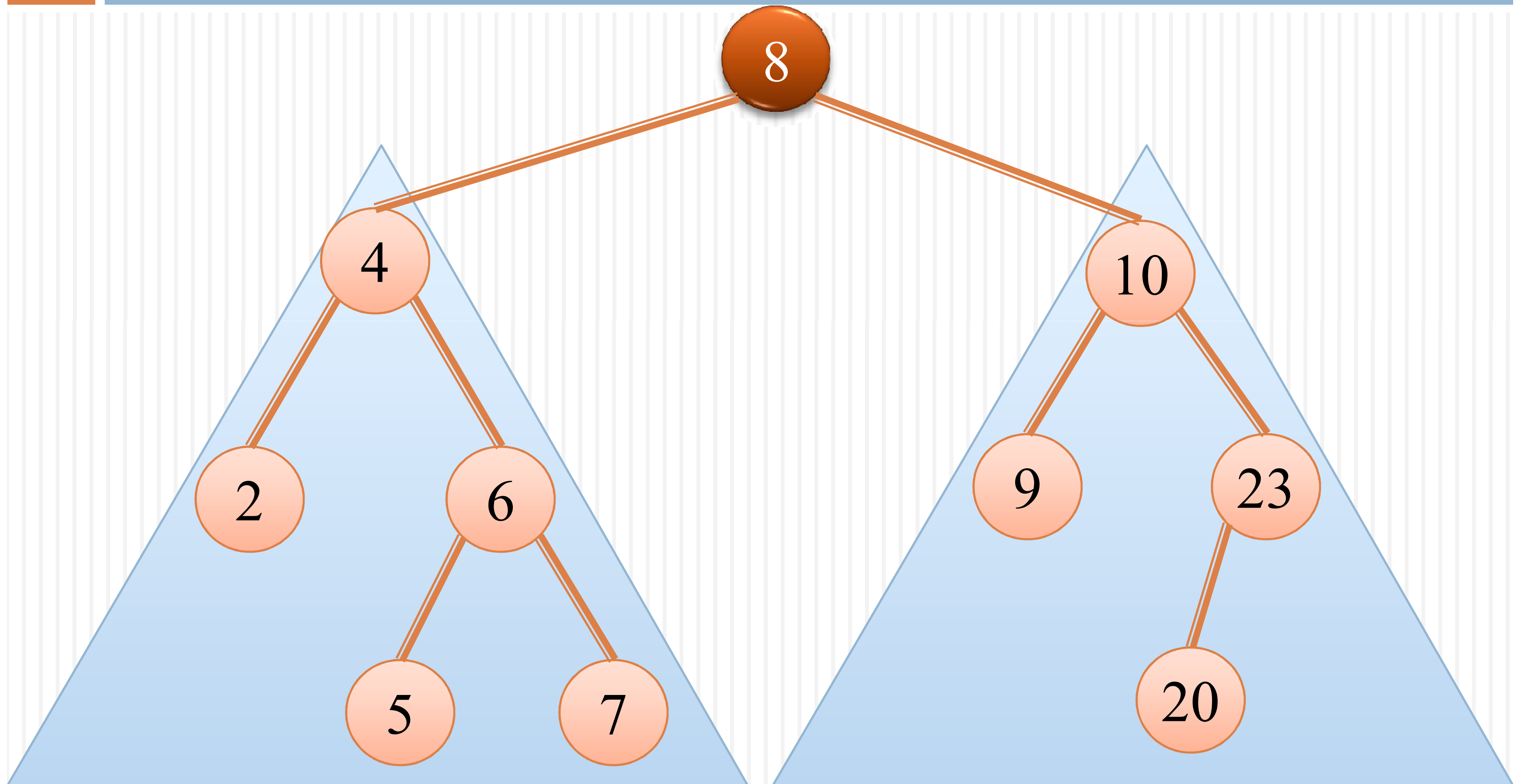
Phân tích cấu trúc
(phân tích cú pháp)



Cây nhị phân tìm kiếm

- Cây nhị phân tìm kiếm là cây nhị phân thỏa mãn các điều kiện sau:
  1. Khóa của các đỉnh thuộc cây con trái nhỏ hơn khóa của gốc
  2. Khóa của các đỉnh thuộc cây con phải lớn hơn khóa của gốc
  3. Cây con trái và cây con phải của gốc cũng là cây nhị phân tìm kiếm

Cây nhị phân tìm kiếm



Cây nhị phân tìm kiếm
Thao tác tìm kiếm

- *Bước 1*: Bắt đầu từ gốc
- *Bước 2*: So sánh dữ liệu cần tìm với dữ liệu của nút hiện hành
  - Nếu bằng nhau ➔ Tìm thấy. Kết thúc
  - Nếu nhỏ hơn ➔ Đi qua nhánh trái. Tiếp bước 2
  - Nếu lớn hơn ➔ Đi qua nhánh phải. Tiếp bước 2
- *Bước 3*: Không thể đi tiếp nữa ➔ Không tìm thấy. Kết thúc
- Thời gian chạy xấu nhất tỉ lệ với chiều cao của cây

Cây nhị phân tìm kiếm
Thao tác tìm kiếm


```
NODE* search(NODE *r, int key)
{
    if(r == NULL)
        return NULL;

    if(key == r->data)
        return r;

    if(key < r->data)
        return search(r->left, key);
    else
        return search(r->right, key);
}
```

```
NODE* search(BTREE *t, int key)
{
    NODE *n = t->root;
    while(n != NULL)
    {
        if(n->data == key)
            return n;

        if(key < n->data)
            n = n->left;
        else
            n = n->right;
    }

    return NULL;
}
```

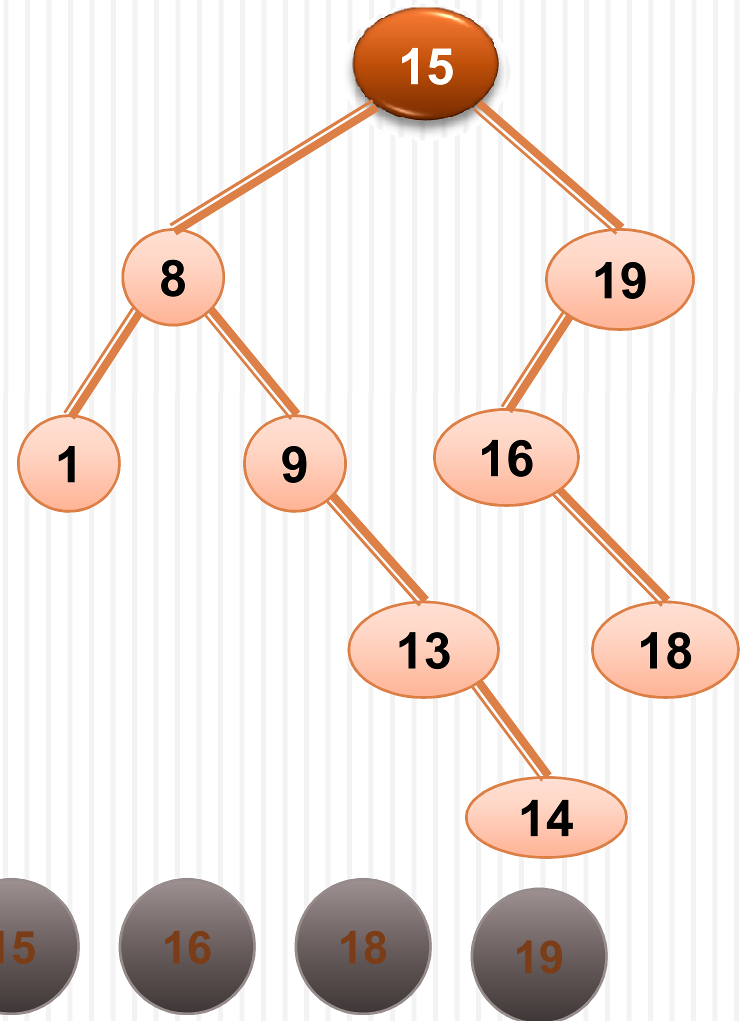
Đệ qui đuôi

Kĩ thuật lặp

Cây nhị phân tìm kiếm
Sắp xếp bằng cây nhị phân tìm kiếm

- *Nhận xét*: phép duyệt giữa trên cây nhị phân tìm kiếm cho thứ tự sắp xếp tăng dần

  ➔ Có thể dùng cây nhị phân tìm kiếm để sắp xếp



1 8 9 13 14 15 16 18 19

31

Hỏi và Đáp